Data Visualization in R

Alark Joshi

Reading a CSV file

- Hotdogs <- read.csv(“contest-winners.csv”, sep =“,”, header = TRUE)

Bar graphs in R

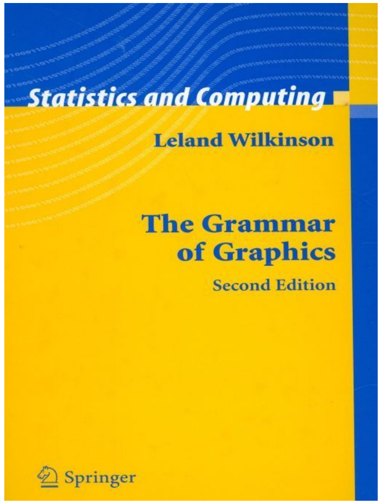
- `barplot(hotdogs$Dogs.eaten)`
- `barplot(hotdogs$Dogs.eaten, names.arg =hotdogs$Year, col=“red”, border = NA, xlab=“Year, ylab =“Hot dogs and buns (HDB) eaten”)`

Scatterplots in R

- plot(population$Year, population $Population, type="1", ylim="c(0,700000000), xlab="Year", ylab="Population)

ggplot2

- `ggplot2` is a data visualization package for the statistical programming language R.
- Created by Hadley Wickham in 2005, ggplot2 is an implementation of Leland Wilkinson's Grammar of Graphics



Statistics and Computing

Leland Wilkinson

The Grammar of Graphics

Second Edition

Springer

ggplot2

- It is a general scheme for data visualization which breaks up graphs into semantic components such as scales and layers.
- ggplot2 can serve as a *replacement for the base graphics in R* and contains a number of defaults for **web** and **print** display of common scales.

ggplot2 datasets

Data sets included in ggplot2 and used in examples

1. `diamonds` - Prices of 50,000 round cut diamonds
2. `economics` - US economic time series.
3. `midwest` - Midwest demographics.
4. `movies` - Movie information and user ratings from IMDB.com.
5. `mpg` - Fuel economy data from 1999 and 2008 for 38 popular models of car
6. `msleep` - An updated and expanded version of the mammals sleep dataset.
7. `presidential` - Terms of 10 presidents from Eisenhower to Bush W.
8. `seals` - Vector field of seal movements.

Scatterplots

- `head(iris) or head(iris, n=10)`
- Let’s plot Sepal.Length against Petal.Length using ggplot2’s qplot() function.

Credits: http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/

Scatterplots

- `qplot(Sepal.Length, Petal.Length, data = iris)`

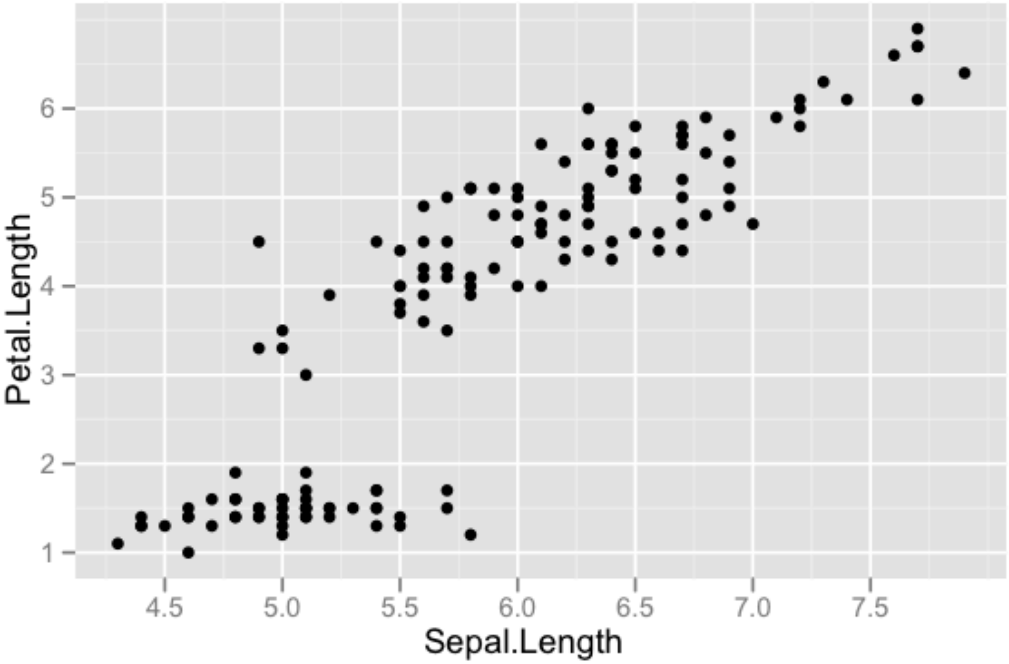


Image Credits: http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/

Scatterplots

- `qplot(Sepal.Length, Petal.Length, data = iris, color = Species)`

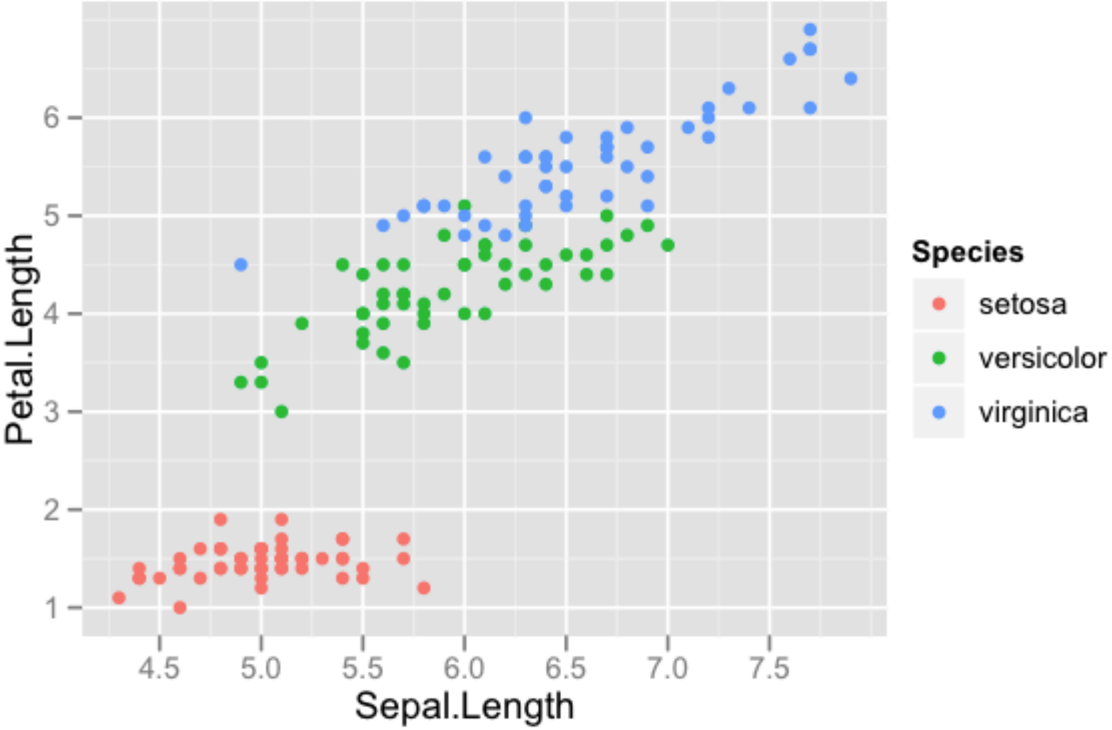


Image Credits: http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/

Scatterplots

- We can let the size of each point denote sepal width, by adding a `size = Sepal.Width` argument.

- `qplot(Sepal.Length, Petal.Length, data = iris, color = Species, size = Petal.Width)`

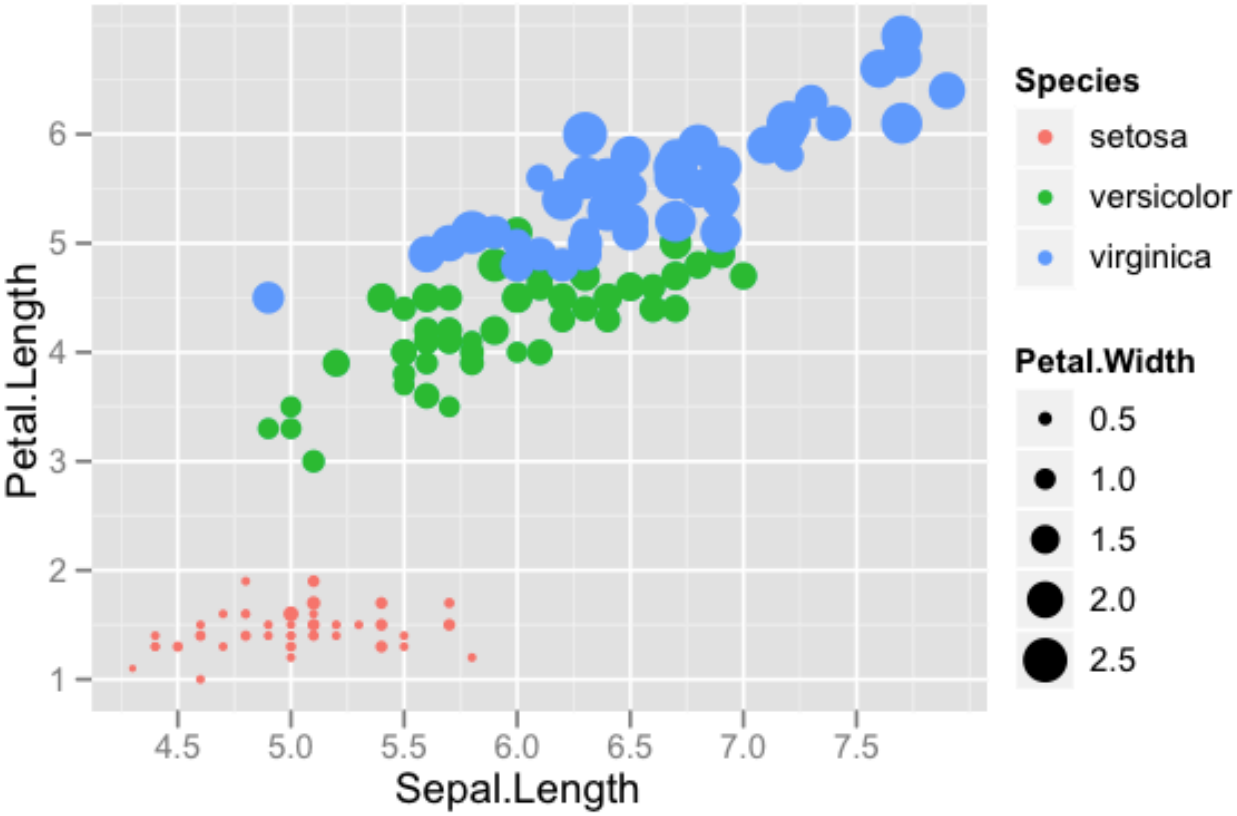


Image Credits: http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/

- `qplot(Sepal.Length, Petal.Length, data = iris, color = Species, size = Petal.Width, alpha = I(0.7))`

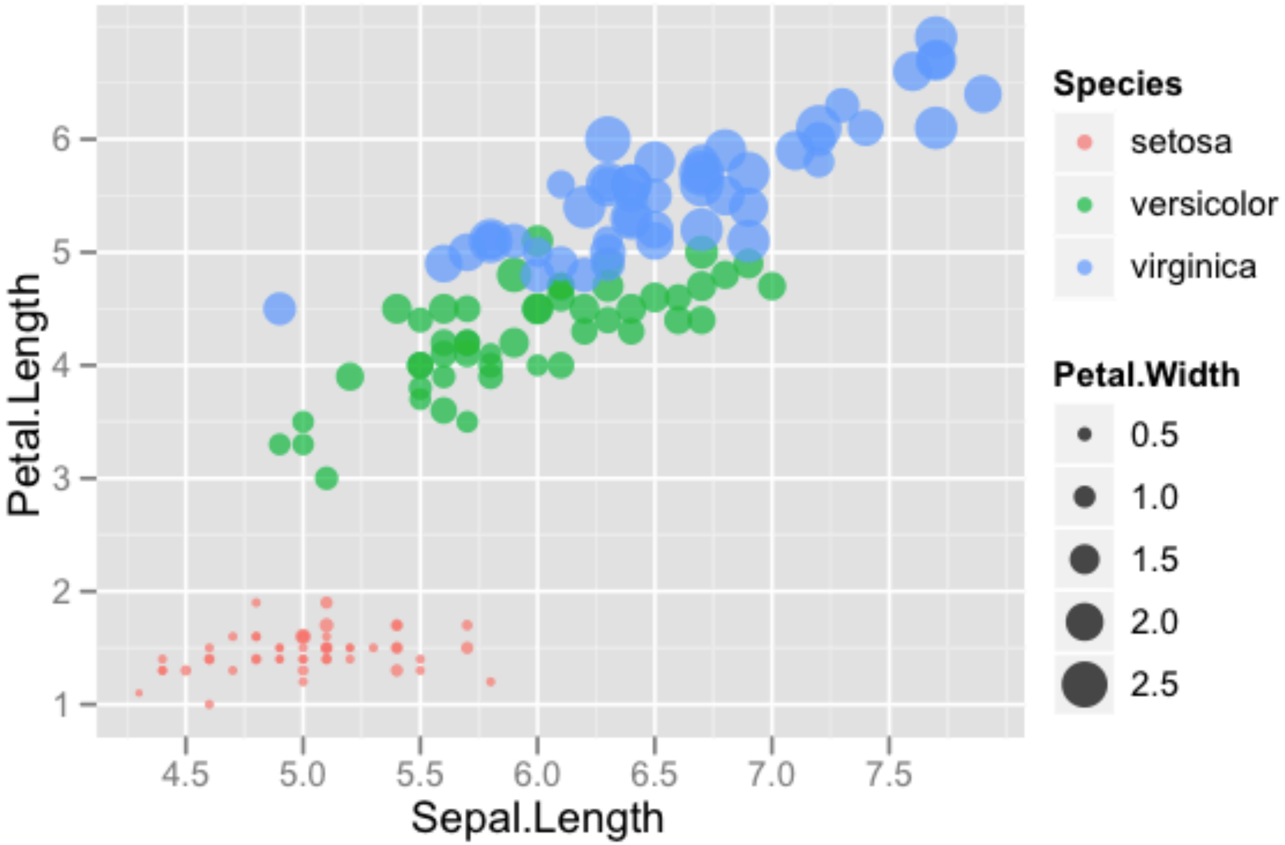


Image Credits: http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/

Other `geom`’s

- We’ve been using a `point` **geom**-etry

```
movies = data.frame(
    director = c("spielberg", "spielberg", "spielberg",
"jackson", "jackson"),
    movie = c("jaws", "avatar", "schindler's list",
"lotr", "king kong"), minutes = c(124, 163, 195, 600,
187))
```

Image Credits: http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/

Bar chart (geom)

- Plot the number of movies each director has.

```
qplot(director, data =
movies, geom = "bar",
ylab = "# movies")
```

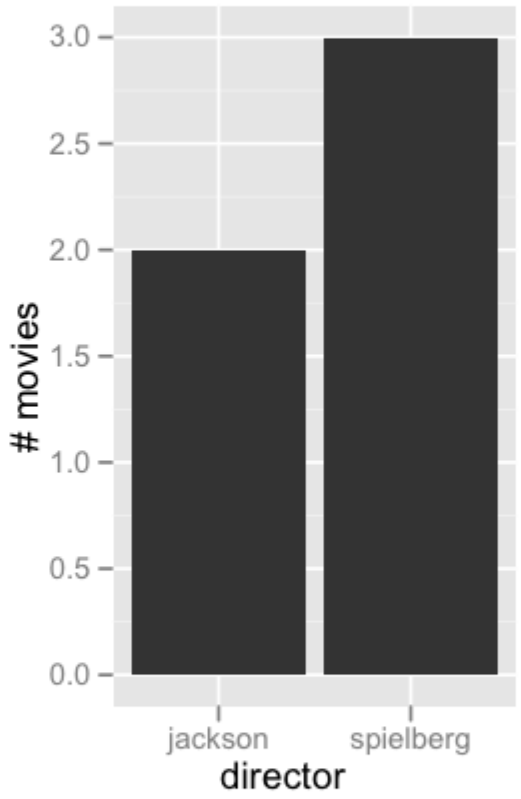


Image Credits: http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/

Line chart (geom)

- \`Orange\` is another built-in data frame

```
qplot(age, circumference, data = Orange,
geom = "line", colour = Tree,   main =
"How does orange tree circumference vary
with age?")
```

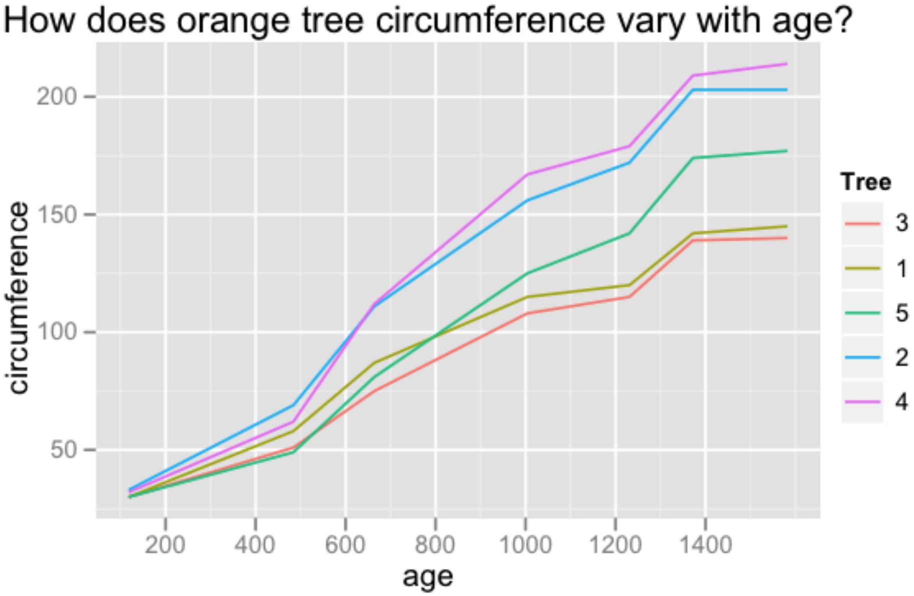


Image Credits: http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/

Plot both points & lines

```
qplot(age, circumference, data =
Orange, geom = c("point", "line"),
colour = Tree)
```

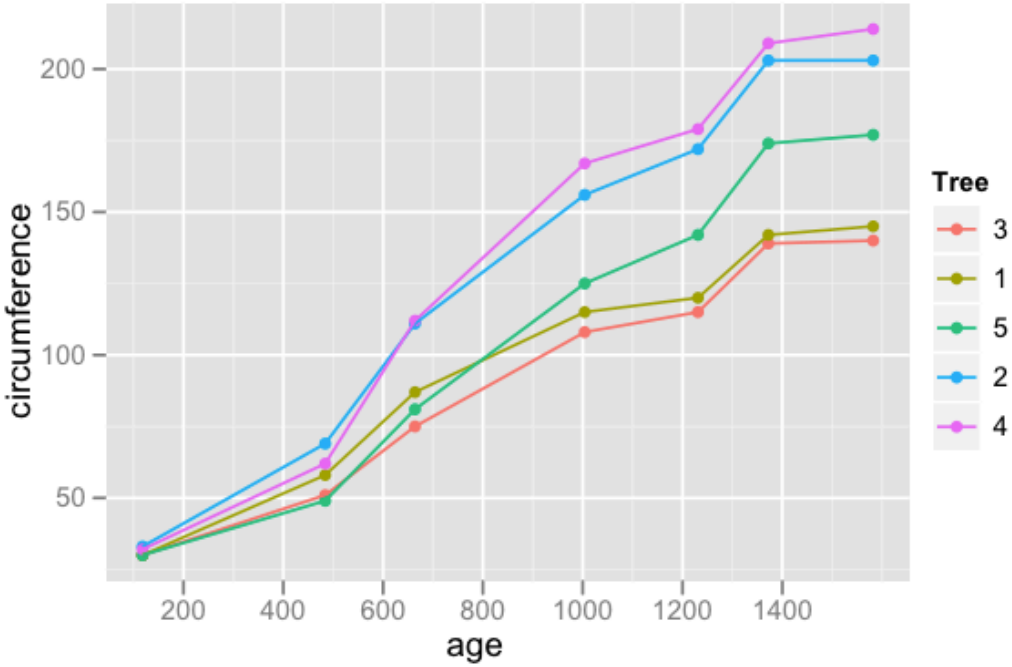
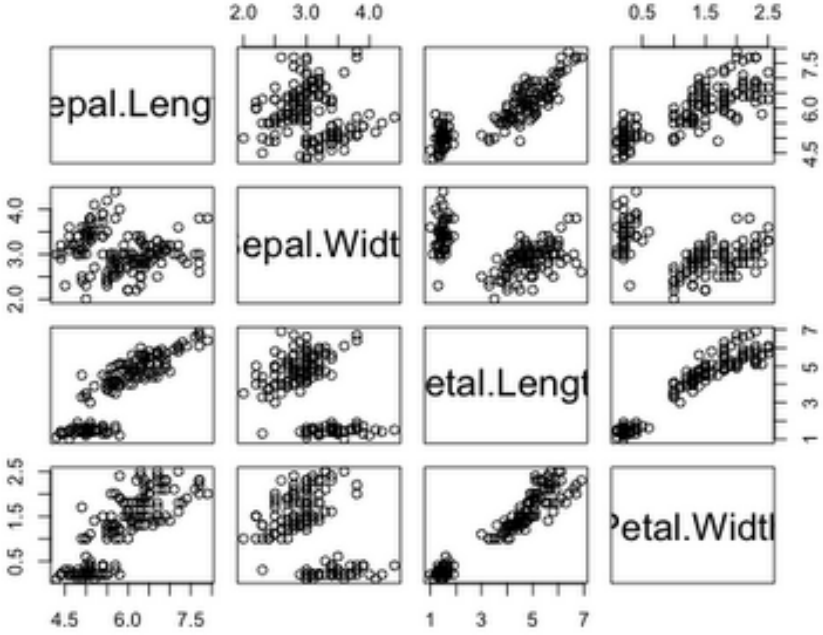


Image Credits: http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/

Scatterplot matrix

1. `require(lattice)`
2. `require(ggplot2)`
3. `pairs(iris[1:4], pch=21)`



Exercise

1. Explore the diamonds dataset (from ggplot2) and find the relationship between carat & price,
   - `qplot(carat, price/carat, data=diamonds)`
2. Plot a histogram of the movies dataset plotting the movie ratings
3. Load a CSV file and explore its relationships

Running a R script

- source(“script.R”)